

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: 2 0 0 4 年 6 月 2 2 日

出 願 番 号
Application Number: 特 願 2 0 0 4 - 1 8 3 9 8 8

パリ条約による外国への出願
に用いる優先権の主張の基礎
となる出願の国コードと出願
番号

The country code and number
of your priority application,
to be used for filing abroad
under the Paris Convention, is

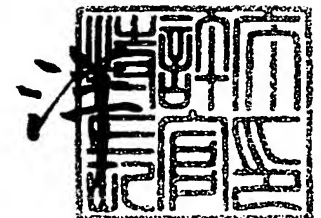
J P 2 0 0 4 - 1 8 3 9 8 8

出 願 人
Applicant(s): 株式会社ソニー・コンピュータエンタテインメント

2 0 0 5 年 5 月 1 1 日

特許庁長官
Commissioner,
Japan Patent Office

小 川



【官 公 庁】 付 託 部
【整理番号】 SCE103034
【提出日】 平成16年 6月22日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 11/00
G06F 15/16
【発明者】
【住所又は居所】 東京都港区南青山2丁目6番21号 株式会社ソニー・コンピュー
ターエンタテインメント内
【氏名】 安達 健一
【発明者】
【住所又は居所】 東京都港区南青山2丁目6番21号 株式会社ソニー・コンピュー
ターエンタテインメント内
【氏名】 矢澤 和明
【発明者】
【住所又は居所】 東京都港区南青山2丁目6番21号 株式会社ソニー・コンピュー
ターエンタテインメント内
【氏名】 瀧口 巖
【発明者】
【住所又は居所】 東京都港区南青山2丁目6番21号 株式会社ソニー・コンピュー
ターエンタテインメント内
【氏名】 今井 敦彦
【発明者】
【住所又は居所】 東京都港区南青山2丁目6番21号 株式会社ソニー・コンピュー
ターエンタテインメント内
【氏名】 田村 哲司
【特許出願人】
【識別番号】 395015319
【氏名又は名称】 株式会社ソニー・コンピュータエンタテインメント
【代理人】
【識別番号】 100105924
【弁理士】
【氏名又は名称】 森下 賢樹
【電話番号】 03-3461-3687
【手数料の表示】
【予納台帳番号】 091329
【納付金額】 16,000円
【提出物件の目録】
【物件名】 特許請求の範囲 1
【物件名】 明細書 1
【物件名】 図面 1
【物件名】 要約書 1

【請求項 1】

プロセッサの内部に設けられた複数の処理ブロックの並列利用度を、温度に応じて切り替えることを特徴とするプロセッサの制御方法。

【請求項 2】

請求項 1 に記載の方法において、前記並列利用度と動作周波数の組合せをプロセッサの温度に応じて切り替えることを特徴とするプロセッサの制御方法。

【請求項 3】

請求項 1 に記載の方法において、タスクごとに定まる前記複数の処理ブロックの並列利用可能数を考慮して、前記タスクを割り振ることを特徴とするプロセッサの制御方法。

【請求項 4】

請求項 1 または 3 に記載の方法において、前記複数の処理ブロックのうち、少なくとも最も低い温度の処理ブロックに、タスクを割り振ることを特徴とするプロセッサの制御方法。

【請求項 5】

プロセッサの内部に設けられた複数の処理ブロックの並列利用度と動作周波数の組合せを所定のテーブルを参照して切り替えることを特徴とするプロセッサの制御方法。

【請求項 6】

請求項 5 に記載の方法において、前記テーブルは前記組合せのそれぞれについて、処理のパフォーマンスを記述していることを特徴とするプロセッサの制御方法。

【請求項 7】

請求項 6 に記載の方法において、プロセッサの温度が所定のしきい値を越えると予測されるとき、または超えたとき、前記組合せの中で現在選択されている組合せより発熱量が低い組合せを検出し、検出された組合せへの切替を行うことを特徴とするプロセッサの制御方法。

【請求項 8】

請求項 7 に記載の方法において、検出された組合せが複数存在する場合、パフォーマンスが最大になる組合せへの切替を行うことを特徴とするプロセッサの制御方法。

【請求項 9】

複数の処理ブロックと、
温度を測定するセンサと、

測定された温度に応じて、前記複数の処理ブロックの並列利用度を切り替える制御部と

を備えることを特徴とするプロセッサ。

【請求項 10】

請求項 9 に記載のプロセッサにおいて、前記制御部は、前記温度に応じて、前記並列利用度と動作周波数の組合せを切り替えることを特徴とするプロセッサ。

【請求項 11】

請求項 9 に記載のプロセッサにおいて、前記制御部は、タスクごとに定まる前記複数の処理ブロックの並列利用可能数を考慮して、前記タスクを割り振ることを特徴とするプロセッサ。

【請求項 12】

請求項 9 または 11 に記載のプロセッサにおいて、前記制御部は、前記複数の処理ブロックのうち、少なくとも最も低い温度の処理ブロックに、タスクを割り振ることを特徴とするプロセッサ。

【請求項 13】

複数の処理ブロックと、

前記複数の処理ブロックの並列利用度と動作周波数の組合せを記述するテーブルと、
前記テーブルを参照し、前記組合せを適宜切り替える制御部と、

を備えることを特徴とするプロセッサ。

【請求項 14】

請求項 13 に記載のプロセッサにおいて、前記テーブルは前記組合せのそれぞれについて、処理のパフォーマンスを記述していることを特徴とするプロセッサ。

【請求項 15】

請求項 14 に記載のプロセッサにおいて、当該プロセッサの温度が所定のしきい値を越えると予測されるとき、または超えたとき、前記制御部は、前記組合せの中で現在よりも発熱量が低い組合せを選択し、その組合せへの切替を行うことを特徴とするプロセッサ。

【請求項 16】

各種のタスクを実行するプロセッサを備える情報処理装置であって、
前記プロセッサは、
複数の処理ブロックと、
温度を測定するセンサと、
測定された温度に応じて、前記複数の処理ブロックの並列利用度を切り替える制御部と

を備えることを特徴とする情報処理装置。

【請求項 17】

請求項 16 に記載の装置において、前記制御部は、前記温度に応じて、前記並列利用度と動作周波数の組合せを切り替えることを特徴とする情報処理装置。

【請求項 18】

請求項 16 に記載の装置において、前記制御部は、タスクごとに定まる前記複数の処理ブロックの並列利用可能数を考慮して、前記タスクを割り振ることを特徴とする情報処理装置。

【請求項 19】

請求項 16 または 18 に記載の装置において、前記制御部は、前記複数の処理ブロックのうち、少なくとも最も低い温度の処理ブロックに、タスクを割り振ることを特徴とする情報処理装置。

【請求項 20】

各種のタスクを実行するプロセッサを備える情報処理装置であって、
前記プロセッサは、
複数の処理ブロックと、
前記複数の処理ブロックの並列利用度と動作周波数の組合せを記述するテーブルと、
前記テーブルを参照し、前記組合せを適宜切り替える制御部と、
を含むことを特徴とする情報処理装置。

【請求項 21】

各種のタスクを実行するプロセッサを備える情報処理システムであって、
前記プロセッサは、
複数の処理ブロックと、
温度を測定するセンサと、
測定された温度に応じて、前記複数の処理ブロックの並列利用度を切り替える制御部と

を備えることを特徴とする情報処理システム。

【請求項 22】

請求項 21 に記載のシステムにおいて、前記制御部は、前記温度に応じて、前記並列利用度と動作周波数の組合せを切り替えることを特徴とする情報処理システム。

【請求項 23】

請求項 21 に記載のシステムにおいて、前記制御部は、タスクごとに定まる前記複数の処理ブロックの並列利用可能数を考慮して、前記タスクを割り振ることを特徴とする情報処理システム。

【請求項 24】

請求項 21 または 23 に記載のシステムにおいて、前記制御部は、前記複数の処理プロ

- ・、 ΔT の ΔT 、 ΔT な、最も低い温度の処理ブロックに、 ΔT を割り振ることを特徴とする情報処理システム。

【請求項 25】

各種のタスクを実行するプロセッサを備える情報処理システムであって、
前記プロセッサは、
複数の処理ブロックと、
前記複数の処理ブロックの並列利用度と動作周波数の組合せを記述するテーブルと、
前記テーブルを参照し、前記組合せを適宜切り替える制御部と、
を含むことを特徴とする情報処理システム。

【請求項 26】

プロセッサの内部に設けられた複数の処理ブロックの並列利用度を、温度に応じて切り替えることを特徴とするプロセッサの制御プログラム。

【請求項 27】

請求項 26 に記載のプログラムにおいて、前記並列利用度と動作周波数の組合せをプロセッサの温度に応じて切り替えることを特徴とするプロセッサの制御プログラム。

【請求項 28】

請求項 26 に記載のプログラムにおいて、タスクごとに定まる前記複数の処理ブロックの並列利用可能数を考慮して、前記タスクを割り振ることを特徴とするプロセッサの制御プログラム。

【請求項 29】

請求項 26 または 28 に記載のプログラムにおいて、前記複数の処理ブロックのうち、少なくとも最も低い温度の処理ブロックに、タスクを割り振ることを特徴とするプロセッサの制御プログラム。

【請求項 30】

プロセッサの内部に設けられた複数の処理ブロックの並列利用度と動作周波数の組合せを所定のテーブルを参照して切り替えることを特徴とするプロセッサの制御プログラム。

【発明の名称】 プロセッサ、情報処理装置およびプロセッサの制御方法

【技術分野】

【0001】

この発明はプロセッサ技術に関し、チップの温度に応じでパフォーマンスを制御するプロセッサ、およびプロセッサの制御方法に関する。

【背景技術】

【0002】

LSI設計において製造プロセスの微細化と素子の高集積化が一段と進み、チップの性能限界として発熱量を考慮することが設計上非常に重要になってきている。チップが高温になると、動作不良を起こしたり、長期信頼性が低下するため、様々な発熱対策がとられている。たとえば、チップの上部に放熱フィンを設けて、チップから発生する熱を逃がす方法がとられる。

【0003】

また、チップの消費電力分布にもとづいて、プロセッサのタスクをスケジューリングすることも検討されている（たとえば、特許文献1参照）。

【特許文献1】 米国特許出願公開第2002/0065049号明細書

【発明の開示】

【発明が解決しようとする課題】

【0004】

急峻な温度上昇が起きた場合等には、上述したような対策では対応しきれない場合もある。そのような場合、チップの動作周波数を下げる等の処理によりパフォーマンス自体を低下させて、対処する手法が考えられる。しかしながら、パフォーマンスを必要以上に低下させることは、無駄である。

【0005】

本発明はこうした課題に鑑みてなされたものであり、その目的は、パフォーマンスの低下を抑えながら、正常動作が保証される範囲に温度を保つことができるプロセッサ、情報処理装置およびプロセッサの制御方法を提供することにある。

【課題を解決するための手段】

【0006】

本発明のある態様はプロセッサの制御方法に関する。この方法は、プロセッサの内部に設けられた複数の処理ブロックの並列利用度を、温度に応じて切り替える。並列利用度と動作周波数の組合せをプロセッサの温度に応じて切り替えてもよい。タスクごとに定まる複数の処理ブロックの並列利用可能数を考慮して、タスクを割り振ってもよいし、複数の処理ブロックのうち、少なくとも最も低い温度の処理ブロックに、タスクを割り振ってもよい。

【0007】

「処理ブロック」は、複数のサブプロセッサを搭載したプロセッサの場合、各サブプロセッサに対応させてもよい。「並列利用度」は、当該サブプロセッサの稼働数に対応させてもよい。

【0008】

本発明の別の態様もプロセッサの制御方法に関する。この方法は、プロセッサの内部に設けられた複数の処理ブロックの並列利用度と動作周波数の組合せを所定のテーブルを参照して切り替える。

【0009】

テーブルは前記組合せのそれぞれについて、処理のパフォーマンスを記述していてもよい。また、プロセッサの温度が所定のしきい値を越えると予測されるとき、または超えたとき、前記組合せの中で現在選択されている組合せより発熱量が低い組合せを検出し、検出された組合せへの切替を行ってもよい。さらに、検出された組合せが複数存在する場合、パフォーマンスが最大になる組合せへの切替を行ってもよい。

【００１０】

本発明のさらに別の態様はプロセッサに関する。このプロセッサは、複数の処理ブロックと、温度を測定するセンサと、測定された温度に応じて、複数の処理ブロックの並列利用度を切り替える制御部と、を備える。制御部は、温度に応じて、並列利用度と動作周波数の組合せを切り替えてもよい。制御部は、タスクごとに定まる前記複数の処理ブロックの並列利用可能数を考慮して、タスクを割り振ってもよいし、複数の処理ブロックのうち、少なくとも最も低い温度の処理ブロックに、タスクを割り振ってもよい。

【００１１】

本発明のさらに別の態様はプロセッサに関する。このプロセッサは、複数の処理ブロックと、本プロセッサの温度を測定するセンサと、測定された温度に応じ、複数の処理ブロックの並列利用度と動作周波数の組合せを切り替える制御部と、を備える。

【００１２】

テーブルは前記組合せのそれぞれについて、処理のパフォーマンスを記述していてもよい。また、プロセッサの温度が所定のしきい値を越えると予測されるとき、または超えたとき、制御部は、前記組合せの中で現在よりも発熱量が低い組合せを選択し、その組合せへの切替を行ってもよい。

【００１３】

本発明のさらに別の態様は情報処理装置に関する。この装置は、各種のタスクを実行するプロセッサを備える情報処理装置であって、プロセッサは、複数の処理ブロックと、温度を測定するセンサと、測定された温度に応じて、複数の処理ブロックの並列利用度を切り替える制御部と、を備える。

【００１４】

なお、以上の構成要素の任意の組合せ、本発明の表現を方法、装置、システム、コンピュータプログラムなどの間で変換したものもまた、本発明の態様として有効である。

【発明の効果】

【００１５】

本発明によれば、パフォーマンスの低下を抑えながら、正常動作が保証される範囲にプロセッサの温度を保つことができる。

【発明を実施するための最良の形態】

【００１６】

（第１実施形態）

図１は、第１実施形態におけるプロセッサの構成を示す図である。プロセッサ１は、チップ内に、メインプロセッサ１００、４個の第１～４サブプロセッサ２００ａ～ｄ、記憶部３００、および温度センサ４００を含む。これらは、図示しないバスにより接続されている。また、サブプロセッサ２００の数は、４個に限るものではなく任意の数を設けることができる。たとえば、８個設けてもよい。また、これらの配置パターンも、図１に示したものに限るものではなく、任意に配置することができる。クロック生成部５００は、プロセッサ１に基本周波数を与える。

【００１７】

メインプロセッサ１００は、プロセッサ１全体を制御する。特に、第１～４サブプロセッサ２００ａ～ｄの管理を行い、これらに適宜タスクを振り分ける。また、温度管理、電力管理、およびパフォーマンス管理を行う。第１～４サブプロセッサ２００ａ～ｄは、メインプロセッサ１００から割り振られたタスクを実行する。温度センサ４００は、測定した温度をメインプロセッサ１００に出力する。温度センサ４００は、パッケージの外に設けてもよいが、急峻な温度変化を測定するために、パッケージ内のダイ（die）に設けるとよい。

【００１８】

図２は、第１実施形態における主にメインプロセッサ１００、記憶部３００および記憶部３００にロードされたソフトウェアの連携によって実現される機能ブロック図である。この機能ブロックがハードウェアとソフトウェアとの組み合わせによっていろいろな形で実

、現行のチップの温度を温度制御部120に出力する。温度制御部120は、温度センサ400から入力される現在の温度と、発熱量推定部110から入力される推定発熱量とを基に、 Δt 期間後の温度を推定する。この関係式を下記(式1)に示す。

【0019】

$$T_t + \Delta t = f(T_t, E) \dots (\text{式1})$$

$T_t + \Delta t$ は Δt 期間後の温度、 T_t は現在の温度、 E は当該 Δt 期間に生じる推定発熱量を示す。このように、当該 Δt 期間後の温度 $T_t + \Delta t$ を、現在の温度 T_t と推定発熱量 E の関数として求める。

【0020】

発熱量推定部110は、サブプロセッサ制御部130から現在稼働しているサブプロセッサ200の数を取得し、周波数制御部140から現在の動作周波数を取得する。そして、それらを基に、推定発熱量 E を求める。この関係式を下記(式2)に示す。

【0021】

$$E = \int [\alpha \cdot C \cdot V_{dd}^2 \cdot f] dt \dots (\text{式2})$$

α は所定の比例定数、 C は負荷を容量で等価的に示した変数、 V_{dd} は電源電圧、 f は動作周波数である。電源電圧 V_{dd} は2乗して用いる。これらを掛け合わせたものを上記 Δt で積分した値が推定発熱量 E となる。

【0022】

本実施形態の負荷の容量 C は、下記(式3)で表される。

$$C = C_m + N C_s \dots (3)$$

C_m はメインプロセッサの容量、 C_s はサブプロセッサ200の容量、 N はサブプロセッサ200の稼働数である。

【0023】

発熱量推定部110は、以上の計算により求めた推定発熱量 E を温度制御部120に渡す。温度制御部120は、上記(式1)に示したように、温度センサ400から取得した現在の温度 T_t と、上記推定発熱量 E とを基に、上記 Δt 期間後の温度 $T_t + \Delta t$ を推定する。そして下記(式4)に示すように、この推定温度 $T_t + \Delta t$ が所定のしきい値温度以上に達したか否かを判断する。所定のしきい値温度は、プロセッサ1全体の正常動作が保証されなくなる温度である。

$$T_t + \Delta t \geq T_{th} \dots (4)$$

【0024】

図3は、動作周波数とサブプロセッサ200の稼働数の組み合わせによる動作ポイントを示す図である。図3において、横軸に遷移可能な動作周波数として、4GHz、2GHz、1GHzを設定している。縦軸に遷移可能なサブプロセッサ200の稼働数として、4個～0個を設定している。0個の場合は、メインプロセッサ100のみが稼働している状態である。したがって、図3は15個の動作ポイントa～oを有する。一番右上の動作ポイントaのパフォーマンスが最も高く、一番左下の動作ポイントoのパフォーマンスが最も低い。通常のフル稼働の場合、動作ポイントaで動作し、サブプロセッサ200の稼働数が4で、動作周波数が4GHzとなる。

【0025】

タスク管理部121は、上記 Δt 期間後のタスクの実行状況を把握し、その時点のサブプロセッサ200の並列利用可能数を特定する。そして、当該並列利用可能数を温度制御部120に渡す。タスクの実行状況には、1つのタスクを実行している場合や複数のタスクを実行している場合がある。また、各タスクの性質により1つのサブプロセッサ200でしか実行できないタスクや、複数のサブプロセッサ200で実行可能なタスクがある。

【0026】

図4は、パフォーマンステーブル122を示す図である。サブプロセッサ200の並列利用数に対する動作ポイントの候補が登録されている。動作ポイントは、上からパフォーマンスが高い順に登録されている。たとえば、サブプロセッサ200を2つ利用する場合

、の動作ポイントは、パフォーマンスへが同じ順に Δt 期間後のサブプロセッサ200の並列利用可能数を取得し、それを基にパフォーマンステーブル122を参照して、遷移する動作ポイントの候補を特定する。サブプロセッサ制御部130は、温度制御部120の指示によりサブプロセッサ200の稼働数を切り替える。周波数制御部140は、温度制御部120の指示により上記動作周波数を切り替える。

【0027】

温度制御部120は、推定温度 $T_t + \Delta t$ が所定のしきい値温度に達した場合、並列利用しているサブプロセッサ200の稼働数を減らすか、プロセッサ1全体の動作周波数を下げて、発熱量を下げなければならない。その際、温度制御部120は、タスク管理部121から入力される上記 Δt 期間後のサブプロセッサ200の並列利用可能数を取得し、それを基にパフォーマンステーブル122を参照して、遷移する動作ポイントの候補を特定する。サブプロセッサ制御部130は、温度制御部120の指示によりサブプロセッサ200の稼働数を切り替える。周波数制御部140は、温度制御部120の指示により上記動作周波数を切り替える。

【0028】

図5は、図2に示した機能ブロックの動作を説明するためのフローチャートである。まず、温度制御部120は、温度センサ400からチップ内の現在の温度を取得する(S10)。次に、発熱量推定部110は、サブプロセッサ制御部130から現在のサブプロセッサ200の稼働数を取得し、周波数制御部140から現在のプロセッサ1全体の動作周波数を取得する。そして、それらを上記(式2)、(式3)に代入して、上記 Δt 期間に生じる発熱量を推定し、温度制御部120に渡す(S11)。温度制御部120は、温度センサ400から取得した現在の温度と、発熱量推定部110により推定された発熱量とを基に、上記 Δt 期間後の温度を推定する(S12)。

【0029】

次に、温度制御部120は、当該推定温度と、所定のしきい値温度とを比較する(S13)。当該推定温度が所定のしきい値温度に達していない場合(S13のN)、上記 Δt 期間後の温度に対する正常な動作が保証されるため、現在のサブプロセッサ200の稼働数および動作周波数を切り替える必要はない。

【0030】

当該推定温度が所定のしきい値温度以上の場合(S13のY)、タスク管理部121から、当該 Δt 期間後のサブプロセッサ200の並列利用可能数を取得する(S14)。そして、パフォーマンステーブル122を参照する(S15)。より具体的には、パフォーマンステーブル122の中の、上記並列利用可能数の項目を参照し、現在の動作ポイントから遷移する次の動作ポイントの候補を特定する。パフォーマンステーブル122は、上記並列利用可能数の項目ごとに、遷移可能な動作ポイントを、パフォーマンスを損なわない順に登録されている。したがって、次の動作ポイントの候補は、現在の動作ポイントのパフォーマンスに対し、パフォーマンスの低下が最も小さい動作ポイントとなる。温度制御部120は、当該動作ポイントのサブプロセッサ200の稼働数とプロセッサ1全体の動作周波数とを取得し、発熱量推定部110に渡す。

【0031】

発熱量推定部110は、温度制御部120から渡されたサブプロセッサ200の稼働数とプロセッサ1全体の動作周波数とを上記(式2)、(式3)に代入して、上記 Δt 期間に生じる発熱量を再度推定し、温度制御部120に返す(S16)。なお、パフォーマンステーブル122に予め各動作ポイントの上記 Δt 期間に生じる発熱量が記述されている場合は、それを用いればよい。温度制御部120は、温度センサ400から取得した温度と、発熱量推定部110により推定された発熱量とを基に、上記 Δt 期間後の温度を再度推定する(S17)。

【0032】

次に、温度制御部120は、当該推定温度と、所定のしきい値温度とを再度比較する(S18)。当該推定温度が所定のしきい値温度に達していない場合(S18のN)、現在の動作ポイントの候補に遷移すれば、上記 Δt 期間後の温度に対する正常な動作が保証される。したがって、温度制御部120は、当該動作ポイントの候補に遷移するために、サ

プロセッサ制御部100に指示してプロセッサ200の稼働数を減らすか、同数制御部140に指示してプロセッサ1全体の動作周波数を下げる。または、その両方を行う(S19)。

【0033】

S18において、当該推定温度が所定のしきい値温度以上の場合(S18のY)、S15に遷移して、再度パフォーマンステーブル122を参照する(S15)。そして、現在の動作ポイントの候補から次の動作ポイントの候補に変更する。以後、S16、S17と上述した処理が繰り返され、S17において推定した温度が、所定のしきい値温度未満になるまで(S18のN)、行う。

【0034】

このように、本実施形態によれば、プロセッサ1の温度が所定のしきい値温度以上になると推定される場合に、サブプロセッサ200の稼働数を減らすか、プロセッサ1全体の動作周波数を下げるかして、これを事前に回避することができる。またその際、パフォーマンステーブル122を参照することによって、最もパフォーマンスを損なわない動作ポイントに遷移することができる。

【0035】

なお、上述した1GHz、2GHz、4GHzといった動作周波数は、チップに与えられる周波数であり、チップ内のメインプロセッサ100およびサブプロセッサ200がこの周波数で動作している訳ではない。動作に有効に活用されている周波数、即ち実効周波数は、上述した周波数より低いものである。この実効周波数は、タスクによって異なる。そこで、上記(式2)により発熱量を推定する際、 f に実効周波数を代入してもよい。タスク管理部121は、現在から上記 Δt 期間後までに実行するタスクを特定し、そのタスクに応じた実効周波数を求める。発熱量推定部110は、その実効周波数を上記(式2)に代入して推定発熱量 E を算出する。これによれば、推定発熱量 E を算出する段階でも、タスクを考慮することができ、より精度の高い温度推定を行うことができる。

【0036】

(第2実施形態)

第2実施形態は、第1実施形態のように推定発熱量 E を計算で求めるのではなく、予めテーブルに登録しておく例である。図6は、第2実施形態における主にメインプロセッサ100、記憶部300および記憶部300にロードされたソフトウェアの連携によって実現される機能ブロック図である。タスクテーブル115は、各タスクごとにサブプロセッサ200の並列利用可能数と発熱量を格納している。図7は、タスクテーブル115を示す図である。タスクの種別としては、たとえばキー入力待ち、MP E Gデータのデコード、音声認識等がある。サブプロセッサ200の並列利用可能数は、タスクによって異なる。メインプロセッサ100のみで実行しなければならないタスクは0となる。並列処理可能なタスクは1以上となる。発熱量は、各タスクごとに上記(式1)における Δt 期間に生じる発熱量である。予め実験的に得た値を登録しておけばよい。

【0037】

タスク管理部121は、上記 Δt 期間のタスクの実行状況を把握して、当該 Δt 期間に実行される1以上のタスクを特定し、発熱量推定部110に渡す。発熱量推定部110は、タスク管理部121により特定されたタスクの種別を基に、タスクテーブル115を参照し、プロセッサ1全体の推定発熱量 E を求める。複数のタスクが実行される場合は、各タスクの発熱量を足せばよい。温度センサ400は、現在のチップの温度を温度制御部120に出力する。

【0038】

温度制御部120は、上記(式1)に示したように、温度センサ400から取得した現在の温度 T_t と、上記推定発熱量 E とを基に、上記 Δt 期間後の温度 $T_t + \Delta t$ を推定する。そして上記(式4)に示すように、この推定温度 $T_t + \Delta t$ が所定のしきい値温度以上に達したか否かを判断する。所定のしきい値温度以上に達した場合、動作ポイントの遷移を行う。動作ポイントの遷移は、第1実施形態において説明したように、図3および図

・ 4に示したパフォーマンスヘッダーフォーマットを参照して行う。

【0039】

タスク管理部121は、上記 Δt 期間後のタスクの実行状況を把握し、その時点のサブプロセッサ200の並列利用可能数を特定する。そして、当該並列利用可能数を温度制御部120に出力する。温度制御部120は、パフォーマンステーブル122を参照して動作ポイントの遷移を行う際、タスク管理部121から取得したサブプロセッサ200の並列利用可能数も考慮する。動作ポイントの遷移に際し、サブプロセッサ制御部130は、温度制御部120の指示によりサブプロセッサ200の稼働数を切り替える。周波数制御部140は、温度制御部120の指示により上記動作周波数を切り替える。

【0040】

なお、図6に示した機能ブロックの動作は、第1実施形態において図5のフローチャートを用いた説明部分と基本的に同様である。S11において、ある上記 Δt 期間に生じる発熱量を推定する際、発熱量推定部110がタスクテーブル115とタスク管理部121から得られるタスクの実行状況を基に発熱量を推定する点が相違する。

【0041】

このように第2実施形態によれば、予めタスクごとの発熱量が記述されたタスクテーブルを参照して、チップ全体の発熱量の推定を行うことにより、タスクを考慮した精度の高い温度推定を簡易な処理で行うことができる。

【0042】

（第3実施形態）

第3実施形態は、温度センサ400をチップ内に複数設ける例である。図8は、第3実施形態におけるプロセッサの構成を示す図である。プロセッサ1は、チップ内に、メインプロセッサ100、4個の第1～4サブプロセッサ200a～d、記憶部300、および4個の第1～4温度センサ400a～dを含む。第1温度センサ400aは、ブロックaの温度を測定し、第2温度センサ400bは、ブロックbの温度を測定する。第3温度センサ400cおよび第4温度センサ400dも同様である。温度センサ400の数は、4個に限るものではなく任意の数を設けることができる。たとえば、2個設けてもよい。また、これらの配置パターンも、図1に示したものに限るものではなく、任意に配置することができる。その他は、図1の説明と同様である。

【0043】

第3実施形態は、図2や図6に示した機能ブロック図と同様の構成で実現可能である。温度センサ400が複数になる点が第1実施形態および第2実施形態と異なる。即ち、温度制御部120には、各ブロックの現在の温度が入力される。

【0044】

図9は、第3実施形態を説明するためのフローチャートである。まず、温度制御部120は、複数の温度センサ400から各ブロックの現在の温度を取得する（S20）。温度制御部120は、それを基に最高温度のブロックを特定する（S21）。次に、発熱量推定部110は、サブプロセッサ制御部130から現在のサブプロセッサ200の稼働数を取得し、周波数制御部140から現在のプロセッサ1の動作周波数を取得する。そして、それらを上記（式2）、（式3）に代入して、上記 Δt 期間に生じる発熱量を推定し、温度制御部120に渡す（S22）。なお、発熱量推定部110は、タスク管理部121により特定されたタスクの種別を基に、タスクテーブル115を参照し、プロセッサ1全体の発熱量を推定してもよい。温度制御部120は、特定した最高温度と、発熱量推定部110により推定された発熱量とを基に、上記 Δt 期間後の温度を推定する（S23）。

【0045】

次に、温度制御部120は、当該推定温度と、所定のしきい値温度とを比較する（S24）。当該推定温度が所定のしきい値温度に達していない場合（S24のN）、上記 Δt 期間後の温度に対する正常な動作が保証されるため、現在のサブプロセッサ200の稼働数および動作周波数を切り替える必要はない。

【0046】

当該推定温度が所定のしきい値温度以上の場合（S24のY）、パフォーマンステーブル121から、当該Δt期間後のサブプロセッサ200の並列利用可能数を取得する（S25）。そして、パフォーマンステーブル122を参照する（S26）。温度制御部120は、パフォーマンステーブル122から遷移する動作ポイントの候補を特定し、当該動作ポイントのサブプロセッサ200の稼働数とプロセッサ1の動作周波数とを取得し、発熱量推定部110に渡す。

【0047】

発熱量推定部110は、温度制御部120から渡されたサブプロセッサ200の稼働数とプロセッサ1の動作周波数とを上記（式2）、（式3）に代入して、上記Δt期間に生じる発熱量を再度推定し、温度制御部120に返す（S27）。温度制御部120は、上記最高温度と、発熱量推定部110により推定された発熱量とを基に、上記Δt期間後の温度を再度推定する（S28）。

【0048】

次に、温度制御部120は、当該推定温度と、所定のしきい値温度とを再度比較する（S29）。当該推定温度が所定のしきい値温度以上の場合（S29のY）、S26に遷移して、再度パフォーマンステーブル122を参照する（S26）。そして、現在の動作ポイントの候補から次の動作ポイントの候補に変更する。以後、S27、S28と上述した処理が繰り返され、ステップS28において推定した温度が、所定のしきい値温度未満になるまで（S29のN）、行う。

【0049】

S29において、上記推定温度が所定のしきい値温度に達していない場合（S29のN）、現在の動作ポイントの候補に遷移すれば、上記Δt期間後の温度に対する正常な動作が保証される。したがって、温度制御部120は、当該動作ポイントに遷移するように、サブプロセッサ制御部130もしくは周波数制御部140、または両方に指示する。まず、当該動作ポイントに遷移するために、サブプロセッサ200の稼働数を削減しなければならない場合（S30のY）、サブプロセッサ制御部130は、温度制御部120から指示されたサブプロセッサ200を停止する（S31）。

【0050】

ここで、温度制御部120は、最高温度のブロックに属しているサブプロセッサ200の停止を指示する。当該ブロックのサブプロセッサ200が停止している場合、当該ブロックに最も近い位置のサブプロセッサ200の停止を指示する。たとえば、図8においてブロックaが最高温度のブロックの場合で、サブプロセッサ200が4個稼働から2個稼働に遷移する場合、第1サブプロセッサ200aと第2サブプロセッサ200bを停止させる。このように、最高温度のブロックに属しているサブプロセッサ200、当該ブロックに近い位置のサブプロセッサ200の順番で停止させていく。また、各ブロックが最高温度になった場合の、サブプロセッサ200を停止させる順番は、予めテーブルに記述しておいてもよい。その場合、温度制御部120は、当該テーブルを参照してサブプロセッサ制御部130に指示する。

【0051】

次に、上記動作ポイントに遷移するために、プロセッサ1全体の動作周波数を変更しなければならない場合（S32のY）、周波数制御部140は、温度制御部120から指示された動作周波数に変更する（S33）。

【0052】

このように第3実施形態によれば、動作ポイントを遷移する際に、最高温度のブロックに属する、または近い位置のサブプロセッサから停止させることにより、チップ内の温度分布を平準化する方向で、チップ内の温度を制御することができる。

【0053】

（第4実施形態）

第4実施形態は、各ブロックの温度に応じてタスクを割り振る例である。第4実施形態におけるプロセッサの構成は、図8の説明と同様である。図10は、第4実施形態におけ

るエにノ1ノノロセツキ100、記憶部000および記憶部000にロードされたノフトウェアの連携によって実現される機能ブロック図である。複数の温度センサ400は、各サブプロセッサ200a～dの現在の温度、またはチップ内の領域を分割して設定した各ブロックa～dの現在の温度を温度制御部120に出力する。複数の温度センサ400の配置は、各サブプロセッサ200a～dの温度を直接測る位置にあってもよく、各ブロックa～dの温度を測る位置にあってもよい。

【0054】

タスク管理部121は、現在のタスクの実行状況や待ち行列状況を把握し、次に実行すべきタスクの並列利用可能数を温度制御部120に渡す。並列利用可能数は、各タスクの性質により異なる。

【0055】

温度制御部120は、複数の温度センサ400から入力される温度を基に、各サブプロセッサ200a～dのテーブルを低温度順に作成する。図11は、第4実施形態における低温度順にソートされたテーブルを示す図である。図11では、温度の低い順に、第4サブプロセッサd→第2サブプロセッサb→第3サブプロセッサc→第1サブプロセッサaとソートされている。この順番は、複数の温度センサ400から入力される温度により、適応的に変更される。なお、このテーブルは、各サブプロセッサ200a～dの温度順番のみならず、各サブプロセッサ200a～dの実際のまたは推定の温度を管理してもよい。

【0056】

複数の温度センサ400から各ブロックのa～dの温度が入力される場合、複数の温度センサ400と、各サブプロセッサ200a～dとの距離関係を基に、計算で各サブプロセッサ200a～dの温度を推定してもよい。

【0057】

温度制御部120は、上記テーブルを参照して、次に実行すべきタスクを低温度のサブプロセッサに割り振る。その際、次に実行すべきタスクの並列利用数を考慮し、割り振るサブプロセッサを決定する。即ち、並列利用度が2の場合、上記テーブルを参照して、低温度順に上から2つのサブプロセッサに当該タスクを割り振る。これによれば、パフォーマンスの低下を抑えながら、温度の平準化を行うことができる。勿論、1つのサブプロセッサにのみタスクを割り振ってもよい。なお、上記テーブルが各サブプロセッサ200a～dの温度も管理している場合、並列利用可能数により割り振り候補となるサブプロセッサであっても、そのサブプロセッサが所定の閾値温度を超える温度であるときは、割り振り候補から除外してもよい。その場合、並列利用可能数より少ないサブプロセッサでタスクを実行することになる。当該閾値温度は、実測やシミュレーションにより最適値を求めて設定するとよいが、割り振るタスクの実行後の温度上昇を加味して、若干低めに設定してもよい。これによれば、スポット的な発熱を抑えることができる。

【0058】

温度制御部120は、タスクを割り振るサブプロセッサをサブプロセッサ制御部130に指示する。サブプロセッサ制御部130は、温度制御部120の指示により各サブプロセッサ200a～dの稼働／非稼働を制御する。

【0059】

図12は、第4実施形態を説明するためのフローチャートである。まず、温度制御部120は、複数の温度センサ400から各サブプロセッサ200a～dの温度、または各ブロックa～dの現在の温度を取得する(S40)。次に、すべてのサブプロセッサ200a～dを低温度順にソートする(S41)。そして、次に実行すべきタスクを低温度順に1つ以上のサブプロセッサに割り振る(S42)。その際、当該タスクにおけるサブプロセッサの並列利用可能数、各サブプロセッサ200a～dの温度が所定の閾値温度を超えないか等を考慮することができる。

【0060】

このように第4実施形態によれば、低温度のサブプロセッサに優先的にタスクを割り振

、つて、複数のサブプロセッサの温度を低下させるように制御することにより、チップ内の温度分布を平準化することができる。また、スポット的な発熱を事前に防止することもできる。なお、本実施形態において、各サブプロセッサの温度を基に、タスクを割り振るサブプロセッサを決定した。この点、各サブプロセッサを、チップ内の領域を分割して設定した各ブロックに読み替えてもよい。この場合、各ブロックの温度を低温度順にソートし、低温度のブロックに近接するサブプロセッサにタスクを割り振ることとなる。

【0061】

以上、本発明を実施形態をもとに説明した。これらの実施形態は例示であり、それらの各構成要素や各処理プロセスの組合せにいろいろな変形例が可能なこと、またそうした変形例も本発明の範囲にあることは当業者に理解されるところである。

【0062】

各実施形態においては、上記 Δt 期間後の推定温度と所定のしきい値温度とを比較した。この点、温度センサ400から取得する現在の温度と所定のしきい値温度とを比較してもよい。この場合、所定のしきい値温度は、各実施形態における温度より低めに設定するとよい。

【0063】

上記(式1)において、上記 Δt 期間後の温度 $T_{t+\Delta t}$ は、現在の温度 T_t と推定発熱量 E との関数であると説明した。この点、サブプロセッサ200のコンビネーションを要素に追加してもよい。上記 Δt 期間後の温度 $T_{t+\Delta t}$ は、主としてタスクの実行状況に依存する。これに加えて、隣接するサブプロセッサ200に同時にタスクが割り振られると、離れたサブプロセッサ200にタスクが分散する場合よりも熱が発生しやすいなどの物理的な性質からの影響も受ける。温度制御部120は、タスクの実行によるサブプロセッサ200のコンビネーションも考慮して、上記 Δt 期間後の温度 $T_{t+\Delta t}$ を推定することができる。

【0064】

第3実施形態において説明したブロックは、スポット的に発熱のピークが現れる領域の大きさに合わせて区切られることが理想であるが、当該ブロックの大きさは、発熱制御の目標精度やプロセッサ1の要求仕様によって自由に決めてよい。また、当該ブロックは同一サイズで規則的に区切られてもよいが、各種プロセッサの境界に合わせて不規則に区切られてもよい。

【0065】

サブプロセッサ200の数、その配置位置、および温度センサ400の数、その配置位置は、任意に設定可能である。また、パフォーマンステーブル122に記述される遷移可能な動作周波数、および遷移可能なサブプロセッサ200の稼働数も、任意に設定可能である。さらに、所定のテーブルに記述されるサブプロセッサ200を停止させる順番も任意に設定可能である。主に最高温度のブロックからの位置により、停止させる順番を定めるが、重要な回路素子からの位置関係も考慮して、順番を定めてもよい。

【0066】

なお、本発明のプロセッサは、PC、ワークステーション、ゲーム機、PDA、携帯電話機等の情報処理装置の制御部に適用可能なことはいうまでもない。また、ネットワーク上に分散された資源を共有してシステムを構築するような形態の情報処理システムにも適用可能である。

【図面の簡単な説明】

【0067】

【図1】第1実施形態におけるプロセッサの構成を示す図である。

【図2】第1実施形態を説明するための機能ブロック図である。

【図3】動作周波数とサブプロセッサの稼働数の組み合わせによる動作ポイントを示す図である。

【図4】パフォーマンステーブルを示す図である。

【図5】図2の機能ブロックの動作を説明するためのフローチャートである。

【図 6】 第 2 実施形態を説明するための機能ブロック図である。

【図 7】 タスクテーブルを示す図である。

【図 8】 第 3 実施形態におけるプロセッサの構成を示す図である。

【図 9】 第 3 実施形態を説明するためのフローチャートである。

【図 10】 第 4 実施形態を説明するための機能ブロック図である。

【図 11】 第 4 実施形態における低温度順にソートされたテーブルを示す図である。

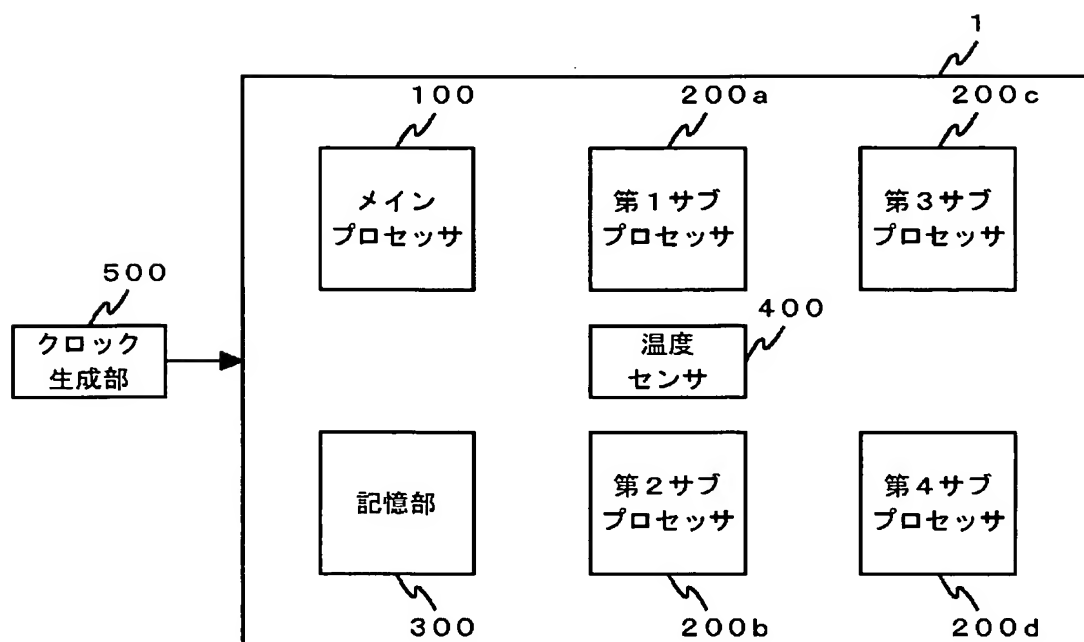
【図 12】 図 10 の機能ブロックを説明するためのフローチャートである。

【符号の説明】

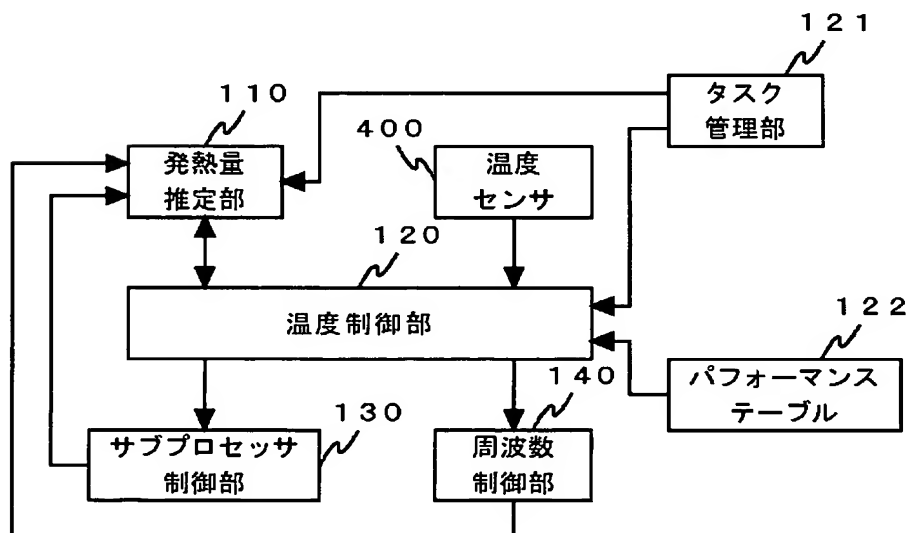
【0068】

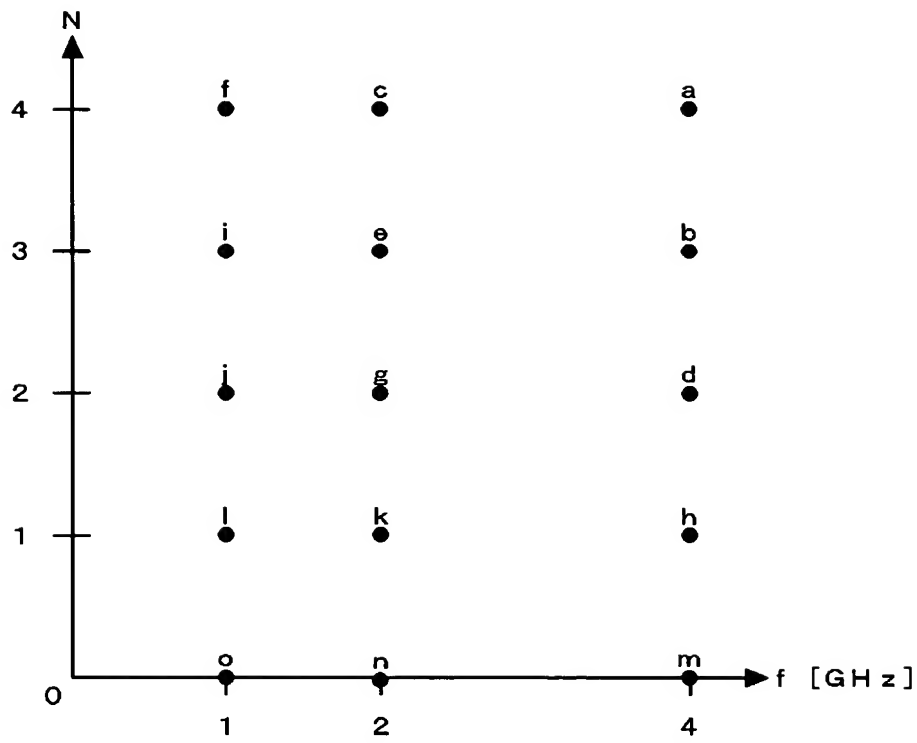
1 プロセッサ、 100 メインプロセッサ、 110 発熱量推定部、 115
タスクテーブル、 120 温度制御部、 121 タスク管理部、 122 パフォー
マンステーブル、 130 サブプロセッサ制御部、 140 周波数制御部、 200
サブプロセッサ、 300 記憶部、 400 温度センサ、 500 クロック生成
部。

【図 1】



【図 2】





【図 4】

サブプロセッサ並列利用数					
パフォーマンス順の動作ポイント	0	1	2	3	4
	m	h	d	b	a
	n	k	g	d	b
	o	l	h	e	c
		m	j	g	d
		n	k	h	e
		o	l	i	f
			m	j	g
			n	k	h
			o	l	i
				m	j
				n	k
				o	l
					m
					n
					o

開始

S 1 0 現在の温度取得

S 1 1 ある時点までの
発熱量を推定

S 1 2 ある時点の温度
を推定

S 1 3 しきい値温度以上？

N

Y

S 1 4 並列利用可能数
を取得

S 1 5 パフォーマンス
テーブル参照

S 1 6 ある時点までの
発熱量を推定

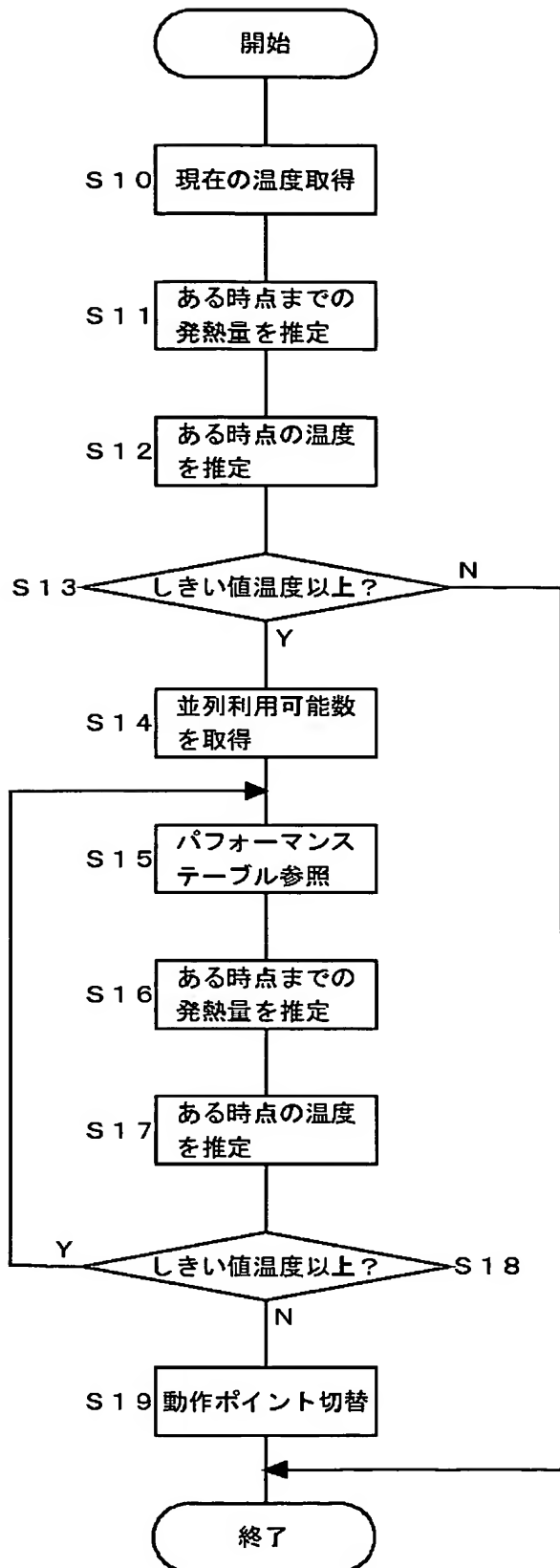
S 1 7 ある時点の温度
を推定

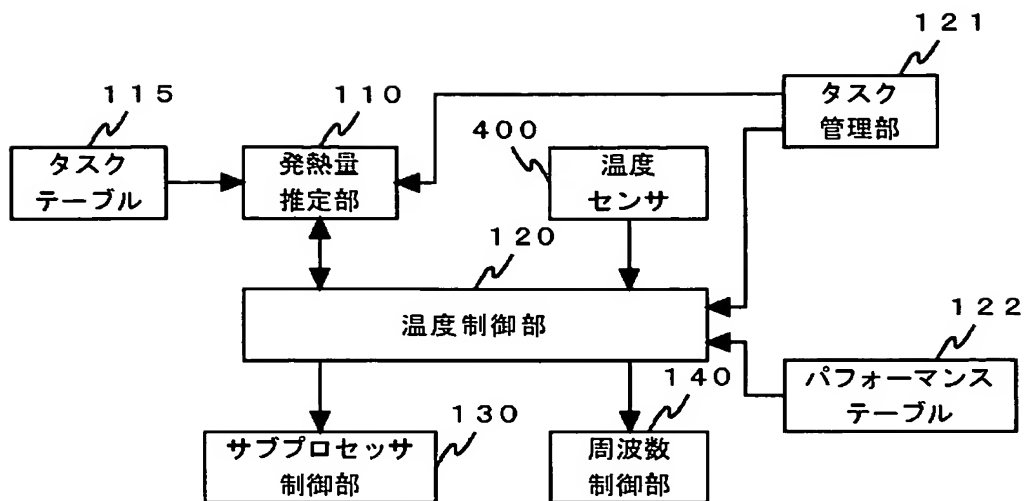
Y しきい値温度以上？ S 1 8

N

S 1 9 動作ポイント切替

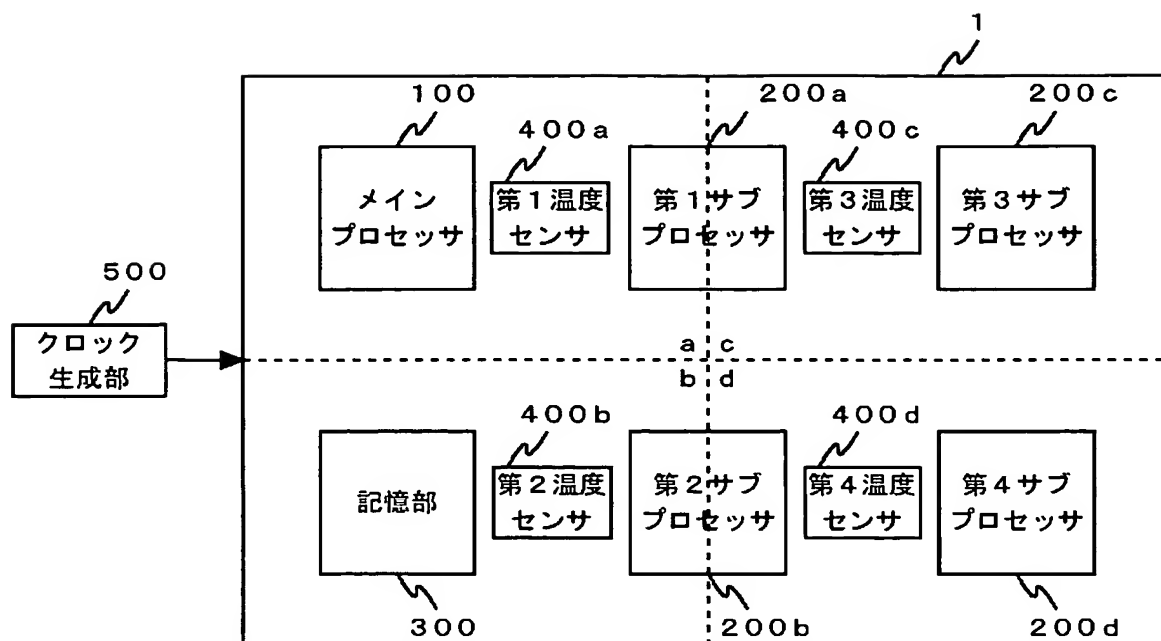
終了

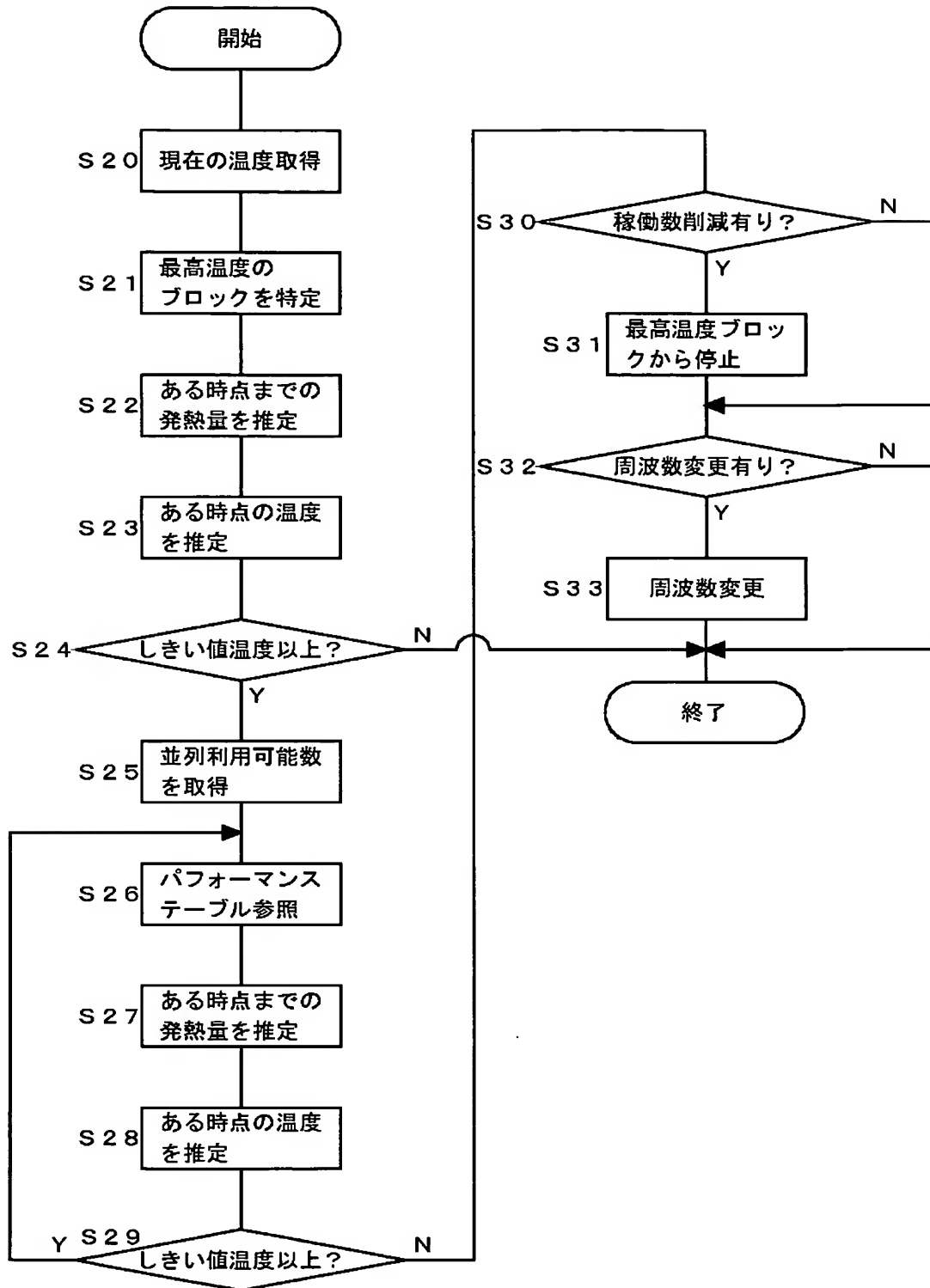


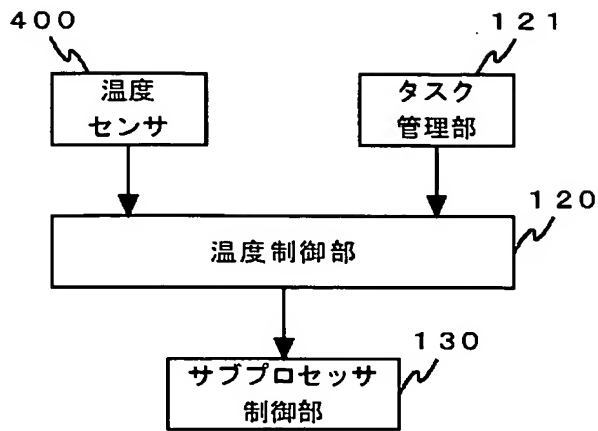


【図 7】

タスク種別	サブプロセッサ 並列利用可能数	発熱量
タスク 1	n 1	E 1
タスク 2	n 2	E 2
タスク 3	n 3	E 3
● ● ●		



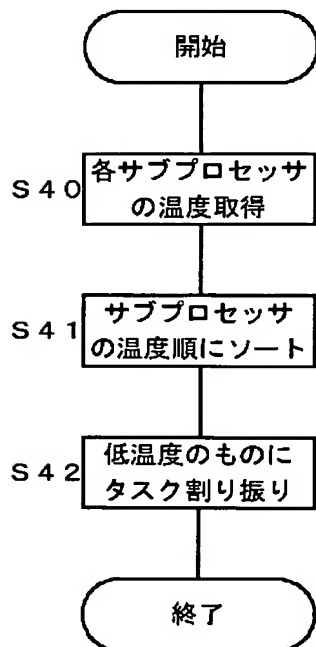




【図 11】

	サブプロセッサ温度
1	第4サブプロセッサ
2	第2サブプロセッサ
3	第3サブプロセッサ
4	第1サブプロセッサ

【図 12】



【要約】

・ 【課題】 パフォーマンスの低下を最小限にしながら、プロセッサの温度を正常動作保証範囲の温度に保つ。

【解決手段】 発熱量推定部 110 は、現在稼働しているサブプロセッサの数を取得し、現在の動作周波数を取得し、 Δt 期間後の発熱量を推定する。温度制御部 120 は、温度センサ 400 から入力される現在の温度と、推定された発熱量とを基に、 Δt 期間後の温度を推定し、所定のしきい値温度と比較する。所定のしきい値温度に達した場合、タスク管理部 121 から Δt 期間後のサブプロセッサの並列利用可能数を取得し、パフォーマンステーブル 122 を参照して遷移する動作ポイントを求める。サブプロセッサ制御部 130 および周波数制御部 140 は、それに応じたサブプロセッサの稼働数と動作周波数に切り替える。パフォーマンステーブル 122 には、遷移する動作ポイントがパフォーマンス順に記述されている。

【選択図】 図 2

【官規石】 予祝補正官
【整理番号】 SCE103034
【提出日】 平成17年 1月17日
【あて先】 特許庁長官殿
【事件の表示】
【出願番号】 特願2004-183988
【補正をする者】
【識別番号】 395015319
【氏名又は名称】 株式会社ソニー・コンピュータエンタテインメント
【代理人】
【識別番号】 100105924
【弁理士】
【氏名又は名称】 森下 賢樹
【電話番号】 03-3461-3687
【手続補正1】
【補正対象書類名】 特許願
【補正対象項目名】 発明者
【補正方法】 変更
【補正の内容】
【発明者】
【住所又は居所】 東京都港区南青山2丁目6番21号 株式会社ソニー・コンピュータエンタテインメント内
【氏名】 安達 健一
【発明者】
【住所又は居所】 東京都港区南青山2丁目6番21号 株式会社ソニー・コンピュータエンタテインメント内
【氏名】 矢澤 和明
【発明者】
【住所又は居所】 東京都港区南青山2丁目6番21号 株式会社ソニー・コンピュータエンタテインメント内
【氏名】 瀧口 巖
【発明者】
【住所又は居所】 東京都港区南青山2丁目6番21号 株式会社ソニー・コンピュータエンタテインメント内
【氏名】 今井 敦彦
【発明者】
【住所又は居所】 東京都港区南青山2丁目6番21号 株式会社ソニー・コンピュータエンタテインメント内
【氏名】 田村 哲司
【その他】 出願の際、発明者中「瀧口 巖」の氏名を「瀧口 巖」と表記した誤記を訂正いたします。

3 9 5 0 1 5 3 1 9

20030701

住所変更

東京都港区南青山二丁目6番21号

株式会社ソニー・コンピュータエンタテインメント

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP05/006965

International filing date: 08 April 2005 (08.04.2005)

Document type: Certified copy of priority document

Document details: Country/Office: JP
Number: 2004-183988
Filing date: 22 June 2004 (22.06.2004)

Date of receipt at the International Bureau: 26 May 2005 (26.05.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse